Bidirectional grammatical encoding using synchronous tree adjoining grammar

Minnen, G.A.G.

Publication date:
1991

Document Version
Publisher's PDF, also known as Version of record

[Link to publication in Tilburg University Research Portal]()

Citation for published version (APA):
Minnen, G. A. G. (1991). *Bidirectional grammatical encoding using synchronous tree adjoining grammar*. (ITK Research Memo). Institute for Language Technology and Artifical Intelligence, Tilburg University.

[redacted]

ITK

MEMO

[redacted]

STITUTE FOR LANGUAGE TECHNOLOGY AND ARTIFICIAL INTELLIGENCE

INSTITUUT VOOR TAAL- EN KENNISTECHNOLOGIE

[redacted]

I.T.K. Research Memo
April 1991

Bidirectional Grammatical Encoding using Synchronous Tree Adjoining Grammar

Guido Minnen

no. 7

Tilburg University, P.O.Box 90153, 5000 LE Tilburg, The Netherlands
Phone: +3113 663113, Fax: +3113 663110.

Bidirectional Grammatical Encoding using Synchronous Tree Adjoining Grammar

Guido Minnen[1]

1 Introduction

Synchronous Tree Adjoining Grammar (TAG), as introduced by Shieber & Schabes ([1990a] and [1990b]), is an extension of Tree Adjoining Grammar (Joshi [1987b]) with respect to semantics. The underlying idea is that the notion, extended with respect to CFG, of the domain of syntactic locality implicit in TAG corresponds to a domain of semantic locality. Therefore TAG can be used for the syntactic, as well as the semantic description of natural language. Shieber & Schabes formalize this idea through a mapping from elementary syntactic TAG structures to their semantic counterparts also stated as elementary TAG structures[2].

The resulting formalism allows for a description of natural language which can be used in a bidirectional fashion, without having to depend on separate architectures for generation and parsing. The way in which this is achieved constitutes a significant improvement upon earlier related approaches to tactical natural language generation (Shieber [1988] and Calder et al. [1989]).

We will investigate whether synchronous TAG satifies the psycholinguistic restrictions imposed upon tactical generation by Levelt's, primarily psycholinguistically oriented, model of the speaker (Levelt [1989]). Aiming at a tactical generator, which is attractive, not only viewed from a computational and linguistic perspective, but also from a psycholinguistic perspective.

[1]I want to thank Gerard Kempen and Erik-Jan van der Linden for valuable discussions and Wietske Sijtsma and Pieter Nieuwint for their comments on earlier drafts of the article.

[2]As Shieber & Schabes point out, synchronous TAG does not necessarily need to constitute a mapping from syntactic to semantic elementary structures; but we will focus on this mapping, because we are primarily interested in the tactical generation of natural language expressions on the basis of logical form expressions.

2 Grammatical Encoding

In order to focus on the restrictions Levelt imposes upon tactical generation, it is necessary to give a short description of Levelt's model of the speaker and the way the separate modules in his model operate with repect to each other.

Levelt distinguishes three independent modules in his model: the *conceptualizer*, the *formulator* and the *articulator*. The conceptualizer can be viewed as the strategic component of the model. The formulator consists of two interdependent components: the *grammatical encoder* and the *phonological encoder*. The grammatical encoder is the component traditionally viewed as the tactical component.

Each of the modules operates on a stream of characteristic input elements. The modules operate on this characteristic input according to Wundt's principle, which states that a module gets activated at the moment it is confronted with a minimal quantity of its characteristic input. Nowhere in the model is there at any time a full representation of the sentence under production; the autonomous modules in the model process a sentence in a parallel piecemeal fashion. Consequently, the modules (and the components within these modules) must operate incrementally.

In order to satisfy the incrementality constraint the grammatical encoder must not only construct tree structures incrementally, but also support incremental realization. In other words, the phonological encoder operates on some pieces of a tree structure without the grammatical encoder having actually finished that tree structure.

The grammatical encoder is the component we are primarily interested in. The characteristic input of this component Levelt calls the preverbal message. A lexical entry (lemma) will be activated when its meaning matches part of the preverbal message. The lemmas are stored in a so-called *mental lexicon*. This lexicon is bidirectional and therefore constitutes a declarative representation of semantic, lexical and phonological information. According to Levelt, the structural syntactic information necessary to construct the tree structure corresponding to the preverbal message is procedurally stored within the grammatical encoder.

We attempt to use the synchronous TAG formalism for grammatical encoding without losing any of the attractive features with respect to bidirectionality mentioned in the preceding section, thereby establishing *bidirectional grammatical encoding*.

3 Synchronous Tree Adjoining Grammar

A synchronous TAG[3] consists of tree pairs and a tree pair consists of two elementary structures, one from the natural language and one from the formal language (logical form).

Nodes, one from each elementary structure in the tree pair, may be linked; we represent such links by means of indices. The interpretation of these links is that operations (substitution or adjunction) on the tree pairs must occur at both ends of a link. After a link has been acted upon, it is removed from the resulting tree structures. All other links are preserved in the result.

When we project the tree pairs in a synchronous TAG onto their first or second components (ignoring the links), the projections are TAGs for a natural language fragment and a logical form fragment, respectively. These grammars are themselves written in a particular variant of TAG; the choice of this *base formalism*, as Shieber & Schabes call it, is free.

As an illustration, a small sample grammar (figure 1) [4], needed for the derivation of sentence (1) is presented.

(1) John kisses Mary tenderly.

Suppose we start with tree pair 1 in figure 1. We choose the link from the subject NP to T and tree pair 3 to perform synchronous substitution to its nodes. Using tree pair 4 on the remaining link from NP to T yields the declarative sentence 'John kisses Mary.'. We can continue the derivation by combining the resultant with tree pair 2, through synchronous adjunction, in order to modify the verb. Figure 2 shows the derived tree pair for the derivation of sample sentence (1).

4 The Lexicalized Multicomponent Base Formalism

4.1 Incorporation

Throughout this article we will use a multicomponent base formalism (Joshi [1987a]). In synchronous multicomponent TAGs the primitive operation is in-

[3] We assume familiarity with previous work on TAGs, throughout the article. See, for instance, the introduction by Joshi [1987a].

[4] We use standard TAG notation, marking foot nodes in auxiliary trees with '$\star$' and nodes where substitution is to occur with '$\downarrow$'. By convention, the set of morphological flexions of a word is written surrounded by backslashes. When a word in a tree is not surrounded by backslashes, it stands for the inflected form. The nonterminal names in the logical form grammar are mnemonic for Formula, Relation (or function) symbol and Term.

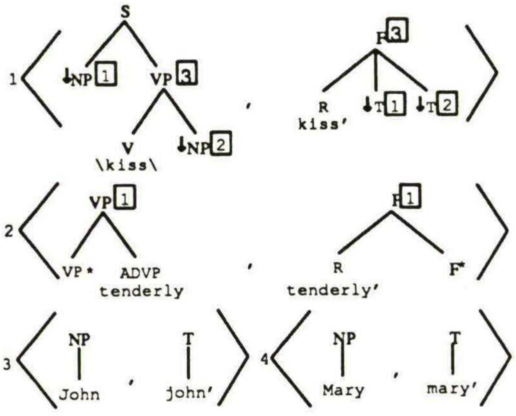

Figure 1

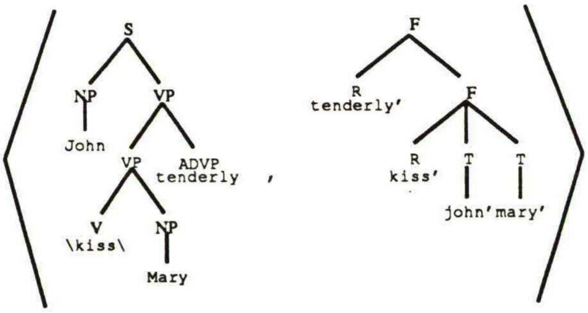

Figure 2

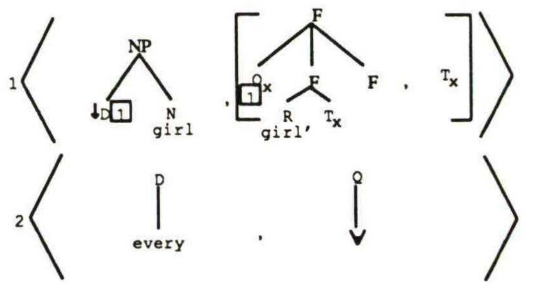

Figure 3

corporation (by multiple substitutions and adjunctions) of a set of elementary structures, at the same time. The links between trees connect a set of nodes in one tree with a set in another. These so-called *hyperlinks*, just like the links introduced in section 3, are represented by means of coindexing. The interpretation of hyperlinks is that when a tree pair is chosen to operate at the link, it must have sets of the correct sizes as its left and right components and the sets are simultaneously used at the various nodes.

The multicomponent base formalism makes it possible to give an elegant account of quantifier scopings (Shieber & Schabes [1990a]). Furthermore, it is a useful extension of TAG in order to describe rightward extraction (extraposition from NP)(Kroch [1987] and Kroch & Joshi [1986]).

A quantifiable noun will be paired with a set of two semantic elementary structures (tree pair 1 in figure 3)[5][6]. This multicomponent tree pair can, for example, be applied to tree pair 1 in figure 1. The determiner can be introduced with the simple tree pair 2.

4.2 Lexicalization

We assume that the tree construction process is lexically guided. This is primarily motivated by arguments put forward by Kempen & Hoenkamp [1987] which are based on lexical idiosyncrasies of verbs with respect to wh-movement and topicalization. In Dutch, for example, the verb *denken* ('to think') behaves differently from the verb *weten* ('to know') with respect to wh-movement over clauses: *denken* does, but *weten* does not, allow wh-movement over clauses. In most rule-based grammar formalisms this results in undesirable non-determinism,

[5] This example is taken from Shieber & Schabes [1990a].

[6] The fact that the right-hand side of tree pair 1 consists of a set of elementary structures is represented by angled bracketing. The subscript x on certain nodes is the value of a feature on the nodes corresponding to the variable bound by the quantifier.

because these lexical idiosyncrasies become available to the tree construction process at a very late stage.

Abeillé [1990] indicates how these lexical idiosyncrasies on the applicability of lexical and syntactic rules can be accounted for in a natural way in TAG by means of lexicalization. In consequence, we will be using a lexicalized version of the multicomponent base formalism, which means that each elementary structure is systematically associated with a lexical item (or semantic terminal).

5 Families of Tree Pairs

The lexicalization of synchronous TAG has far-reaching consequences for the number of tree pairs in the grammar (or rather the lexicon). A lot of lexical items will have a variety of tree pairs. The transitive verb *eat*, for example, requires separate tree pairs for the following constructions: declarative sentence, relativization of the subject, relativization of the object, wh-movement of the subject, wh-movement of the object, topicalization of ..... etc. The members of these so-called *families of tree pairs* (Abeillé & Schabes [1989]) in the lexicon are not formally related[7]. This is unsatisfactory, because one does not want to stipulate the results of syntactic processes, but to describe these processes in such a way that the results follow from them. One cannot be content with a linguistic theory that treats these tree pairs as unrelated.

From a computational point of view, using families of tree pairs is unattractive, because in constructing a tree structure one is forced to solve problems one would rather avoid or postpone. Furthermore, it necessitates a splitting up of the lexicon into families of trees and lexical entries in order to keep it manageable. Although in a few articles on TAG a treatment of families of trees by means of lexical (meta)rules and transformational rules was hinted at (Kroch [1987], Joshi [1987] and Schabes [1990]), this has not been materialized.

The incorporation operation allows a substantial reduction of the size of families of tree pairs. This reduction is obtained by moving structural information associated with the tree pairs in a family to other tree pairs in the lexicon in a way reminiscent of type-lifting in categorial grammar. It is possible to derive, for example, topicalized elementary structures from the elementary structure corresponding to the minimal declarative sentence. This is illustrated in figure 4.

Tree pair 1, representing the minimal declarative sentence with the verb *kiss* as the lexicalized element, can combine with tree pair 2 in case of topicalization[8][9].

[7]Note that this problem is not a result of the mapping between syntax and semantics in synchronous TAG, but that it is a general problem for lexicalized TAG.

[8]The fact that a term is topicalized is represented by means of the terminal TOPIC.

[9]The fact that tree pair 2 establishes topicalization of the object-position remains implicit due to the indices associated with the elementary structures in tree pair 1. In order to allow

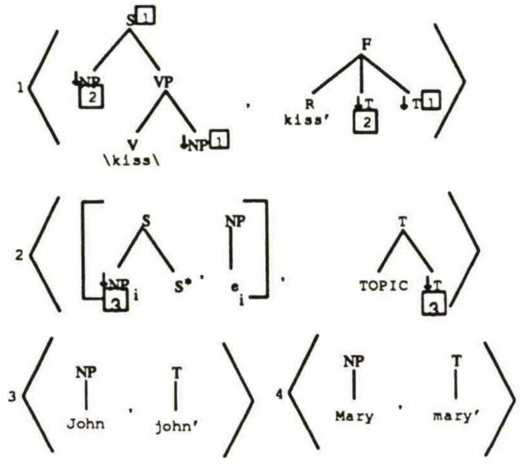

Figure 4

The derived tree pair corresponding to sentence (2) is represented in figure 5.

(2) Mary, John kisses _ .

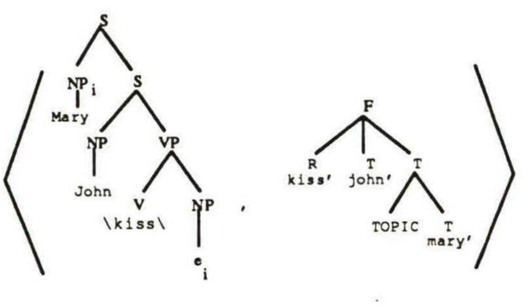

Figure 5

As a result of the description of topicalization using incorporation within the family of tree pairs associated with a transitive verb, which allows topicalization, topicalized structures do not need to be explicitly enumerated[10]. Relativization

topicalization of the subject additional indices are necessary.

[10]Note that it is impossible to lexicalize the left component of tree pair 2. Although this is not obligatory the right component is lexicalized. It seems reasonable to assume that the fact that a term is topicalized is represented in the logical form.

and wh-movement can be treated in a way similar to the treatment of topicalized structures shown above. Let us illustrate the derivation of wh-questions (figure 6).

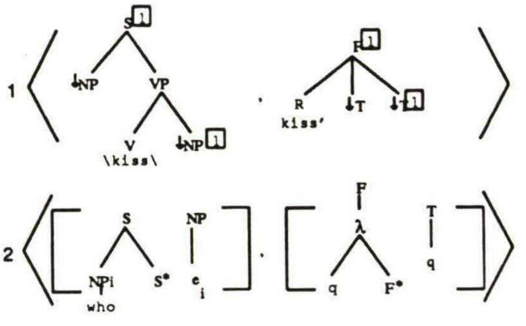

Figure 6

Again, tree pair 1 corresponds to the minimal declarative sentence. In order to derive a wh-question, tree pair 2 has to be adjoined to tree pair 1. The resulting derived tree pair no longer needs to be stated in the lexicon separately.

Tree pair 1 in figure 4 and tree pair 1 in figure 6 are in fact the same, but we have included it again because it needs additional coindexing in order to derive wh-questions. In the lexicon these tree pairs are collapsed into one, and coindexing encodes the possibilities for topicalization, relativization and wh-movement either by means of the number of indices in the right and left component (as noted in 4.1, in order to operate upon a link a tree pair must have the correct sizes as its left and right components) or local constraints upon these indices, which are necessary in order to avoid overgeneration and to justify lexical idiosyncrasies (see section 6.1).

The treatment of topicalization, relativization and wh-movement suggested above changes neither the weak generative capacity (string sets), nor the strong generative capacity (tree sets), because incorporation is only used to perform multiple adjunction and/or substitution into distinct nodes of a single elementary structure (Shieber & Schabes [1990a] and Schabes [1990]). This also prevents the violation of the intuitions about the domain of locality implicit in (synchronous) TAG.

It is of course encouraging that it is possible to reduce the size of families of tree pairs to this extent within a multicomponent base formalism, without violating any of the assumptions underlying (synchronous) TAG. We think that the theoretical and practical advantages make it worth to investigate whether it is possible to take this approach to the limit: minimalizing the structural information associated with the verbs in the lexicon to the minimal declarative structure.

6 Incremental Realization

Like TAG, synchronous TAG allows incremental tree construction. However, incremental construction of tree structures does not ensure that the tactical generation process, as a whole, meets the incrementality constraint (Levelt [1989], Kempen & Hoenkamp [1987]). Due to the fact that the chronological order in which the various elementary structures are attached to the syntactic tree structure need not be identical to their left-to-right order in the utterance, a true incremental tactical generation process must allow the incremental realization of an utterance, (partially) during the construction of the corresponding tree structure (incremental realization). Otherwise, one is forced to postpone the factual realization of an utterance until the tree structure is completed, which means that the ability to construct a tree structure incrementally has only some practical significance.

6.1 Local Constraints on Adjunction

For linguistic descriptions it is convenient and sometimes necessary to be specific as to which elementary structures can be adjoined at a given node; this is especially necessary to avoid overgeneration. This is exactly what is achieved by so-called *local constraints on adjunction* (Joshi [1985],[1987b]). In TAG one can, for each node in an elementary structure, specify one of the following three constraints on adjunction:

- *Selective Adjunction* (SA): Only a specified subset of the set of all elementary structures is adjoinable at node n.
- *Null Adjunction* (NA): No elementary structures are adjoinable at node n.
- *Obligatory Adjunction* (OA): At least one elementary structure (of all the elementary trees adjoinable at n) must be adjoined at n.

Although Shieber & Schabes do not mention this in their description of synchronous TAG, it is obvious that like TAG, synchronous TAG needs to be extended with the possibility of stating local constraints on adjunction. Instead of associating local constraints with nodes in elementary structures, in synchronous TAG, these constraints should, of course, be associated with (hyper)links. For instance, in the case of topicalization the necessity of these constraints becomes apparent. The adjunction of tree pair 2 into tree pair 1 in figure 4 is permitted just once. This cannot be expressed in synchronous TAG without constraining the links in tree pair 2, by means of an NA constraint. More than one elementary structure can be adjoined into another elementary structure as long as each tree is adjoined at a distinct node. Therefore this constraint cannot be imposed upon tree pair 1.

In order to start realizing an utterance, it is necessary to be able to decide whether a certain subtree is finished or not. While this might be trivial in case of CFGs (a subtree is finished if all its non-terminal nodes are expanded up to terminal level), it is not in the case of TAGs, due to the adjunction operation. The adjunction operation allows the expansion of a tree structure even if all its non-terminal nodes have been expanded up to terminal level; this because the adjunction operation allows the insertion of additional structure at any level at any time during the tree construction process.

This means that it is impossible to start realizing a tree structure, without some additional information with respect to which subtrees of the tree under construction are finished and which are not. Although they were not devised for this purpose, local constraints on adjunction provide exactly this essential information, because they specify which adjunctions are to be expected, forbidden or necessary.

The information provided by the NA constraint, supports incremental realization in the most straightforward way. When a link is associated with an NA constraint, the subtrees underneath the nodes that are connected by means of the link are finished, provided it contains no nodes constrained by an SA or OA, or only terminal nodes[11]. It can be concluded that these subtrees are finished and therefore can be realized, whether or not the rest of the tree structure is still under construction.

In the case of a link constrainted by OA, it is evident that the subtrees underneath the nodes connected by the link are not finished. Thus we have conclusive evidence that the factual realization of these subtrees has to be postponed. The information provided by an SA constraint is less straightforward, although not less useful. The fact that a certain adjunction is expected (or rather, admitted) can be used, because it is conceivable that the tree pairs activated by the logical form expression are already available, at least known to be activated[12].

6.2 Factoring Linear Precedence

A time adjunct like _yesterday_ can be adjoined to a derived tree pair corresponding to the sentence 'John kisses Mary' at sentence level. In case, the tree pair corresponding to _yesterday_ is known to be activated, but not yet available, one has to postpone the realization of the entire tree structure, until this tree pair can be used to perform the adjunction. However, the realization of the sentence need not be postponed, if the time adjunct is allowed to occupy sentence final position, like in the case of _yesterday_.

Due to lexicalization, the fact that the time adjunct _yesterday_ can occupy sentence initial and sentence final position, necessitates its inclusion in the lexi-

[11] All possible substitutions within a subtree have to be performed.

[12] Algorithms presented for lexicalized TAG parsing make use of a preselected set of elementary structures, which are activated by the input string (Schabes & Joshi [1991] and Schabes [1990]).

con twice. If we assume a factorization of linear precedence within the syntactic component of the tree pairs in the lexicon, as suggested by Joshi ([1987])[13], these tree pairs can be collapsed into one.

The fact that the time adjunct becomes available at a later stage, can now be reflected by starting the realization of the sentence immediately and thereby ruling out the possibility of *yesterday* occupying initial position, automatically. This way *yesterday* is forced to occupy final position. In case it is not possible for the time adjunct to occupy final position, there is still the possibility of skipping the part of the logical form expression corresponding to the time adjunct or rather start all over again. This way factoring linear precedence aids in avoiding choice problems, thereby enabling the tactical generator to reflect temporal aspects of the generation process in a similar way as in Kempen & Hoenkamp [1987] and De Smedt [1990].

Joshi's ID/LP treatment of Finnish word order (Joshi [1987]) results in tree structures with crossing branches. Although crossing branches can be very useful to describe local discontinuities, it has serious consequences with respect to the computational properties of (synchronous) TAG. It seems feasible to extend Earley's algorithm along the lines suggested by Shieber [1983] in order to allow for direct parsing of ID/LP TAGs; but in case the ID/LP format used results in crossing branches, it becomes necessary to modify the formal definition of tree structures and as a result of that parsing complexity increases (Bunt [1991]).

So far we have not encountered any reason to factor linear precedence in the logical form component of the tree pairs, although this might be helpful to capture certain cases of intentional equivalence, necessary to avoid the problem of logical form equivalence (Appelt [1987] and Shieber [1988]).

7 Bidirectional Grammatical Encoding using Synchronous Tree Adjoining Grammar

Synchronous TAG supports incremental construction of tree structure. Moreover, local constraints on adjunction provide the information necessary to perform realization in an incremental fashion. The factorization of linear precedence makes a mimicking of certain temporal aspects of the generation process possible.

The semantic terminals can be viewed as the minimal characteristic input of the grammatical encoder. The fact that a lemma will be activated when its meaning matches part of the preverbal message can be viewed as retrieving tree pairs from the lexicon as a result of parsing a string of semantic terminals.

[13] There are two differences between the ID/LP format of GPSG and TAG. First, the domain of locality is the elementary structure and second LP restrictions are defined for each elementary structure independently.

The semantic part of the lemma not present in the preverbal message is the structural semantic information associated with the semantic terminal, which becomes available at the moment a tree pair (lemma) will be selected (activated).

Viewing the input of the grammatical encoder as a stream of characteristic input elements corresponds directly to left-to-right Earley parsing of synchronous TAG as suggested by Schabes & Joshi [1988] and Schabes [1990]. The fact that the Earley algorithm does not make use of backtracking, but pursuits alternatives in parallel, even reinforces this view. In addition, the conceptualizer does not need to have any information whether its output corresponds to a complete tree structure; this because Earley's algorithm can determine this on the basis of the tree pairs in the lexicon, just as it does in the case of parsing.

The only property of a synchronous TAG with a lexicalized multicomponent base formalism that does not correspond to Levelt's restrictions on grammatical encoding is the fact that structural syntactic information is also stored in the mental lexicon declaratively, as noted in section 3. This is exactly what we want, because we are interested in bidirectional grammatical encoding and we think that it can be viewed as an important extension of the model with repect to parsing.

As a consequence of the compatibility of synchronous TAG and the grammatical encoder in Levelt's model of the speaker, the question arises how synchronous TAG compares with alternative formalisms that satisfy Levelt's restictions on grammatical encoding. There are, in fact, only two formalisms that do so: Incremental Procedural Grammar (IPG)(Kempen & Hoenkamp [1987]), the grammar formalism Levelt uses in his model, and Segment Grammar (SG) (Kempen [1987] and De Smedt [1990]).

The intuitions with repect to the domain of syntactic locality implicit in synchronous TAG are not shared by either IPG or SG. It is therefore difficult to compare these formalisms. Yet it is possible to point out some important differences between these grammar formalisms

IPG is a formalism in which linguistic knowledge is represented in a fully procedural way. It is therefore impossible to use it for parsing. Within SG linguistic knowledge and control knowledge are separated[14]. However, there are some serious problems to be solved before SG can be used for parsing as well as generation without having to rely on separate architectures (De Smedt 1990]).

In contrast with IPG and SG, synchronous TAG establishes a straightforward mapping between syntax and semantics, which allows an elegant account of quantifier scopings. Furthermore, it is not necessary to use absolute positions in order to obtain an incremental realization of tree structures; this because the local constraints on adjunction provide enough information to distinguish subtrees in the tree structure that are finished and which terminal nodes can be linearized with respect to each other and other lexical items already realized.

[14]SG is a procedural grammar only in the sense that it articulates assumptions about both the format of grammar rules and the structure of the syntactic processor (De Smedt [1990]).

8 Conclusion

We have presented a treatment of topicalization, relativization and wh-movement, thereby bringing about a substantial reduction of the size of families of tree pairs. The base formalism we use, lexicalized multicomponent TAG does not need to be extended, so that the attractive properties of the formalism with repect to reversibility remain applicable. Furthermore, the basic intuition about the domain of locality implicit in (synchronous) TAG is not violated in any way, due to the fact that a restricted version of incorporation is used. We think that it is interesting to investigate whether it is possible to take this approach to the limit by minimalizing the structural information associated with the verbs in the lexicon to the minimal declarative structure (eventually by means of extending the base formalism). This way, it is possible to mimic certain syntactic processes in an elegant and efficient fashion.

The local constraints on (hyper)links associated with the elementary structures in tree pairs in a synchronous TAG (in combination with earlier suggestions with respect to parsing of lexicalized TAG) support incremental realization in a natural way. In the case of factorization of linear precedence it is possible to account for temporal aspects of the tactical generation process.

This incremental realization is a key feature with respect to the compatibility of the synchronous TAG formalism with the psycholinguistic restrictions on natural language production as formulated by Levelt in his model of the speaker. It is argued that left-to-right Earley parsing of logical form expressions does not conflict with Levelt's adoption of Wundt's principle and as a consequence synchronous TAG is very well suitable to be used for grammatical encoding. In fact, it improves upon other formalisms satisfying the psycholinguistic restrictions the model imposes upon grammatical encoding in that it allows bidirectional usage according to a uniform architecture.

We think that bidirectional grammatical encoding using synchronous TAG in the way suggested is attractive not only viewed from a computational and linguistic point of view, but also form a psycholinguistic point of view. At this moment, it is not clear which phenomena allow a description in terms of multicomponent lexicalized tree pairs and which do not. This needs to be subjected to further research. In addition to that, further research is needed into the possibilities of the use of local constraints on substitution, adjunction and incorporation in order to support incremental realization which, of course, heavily depends upon the lexicon being used. The best way to research these issues is probably by implementing a natural language processing module and developing a sizeable lexicon along the lines suggested.

9 References

Abeillé, A [1990], Lexical constraints on Syntactic Rules in a Tree Adjoining Grammar. In *Proceedings of the 28th Meeting of the Association for Computational Linguistics*, Pittsburgh.

Abeillé, A & Schabes, Y [1989], Parsing Idioms in Lexicalized TAGs. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics*, University of Manchester.

Appelt, D [1987] Bidirectional Grammars and the Design of Natural Language Generation Systems. In Y. Wilks *Theoretical Issues in Natural Language Processing-3*, New Mexico State University, New Mexico.

Bunt, H [1991] Parsing with Discontinuous Phrase Structure. In M. Tomita (ed.) *Current Issues in Parsing Technology*, Kluwer Academic Press.

Calder, J, Reape, M & Zeevat, H [1989] An algorithm for generation in Unification Categorial Grammar. In *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics*, University of Manchester Institute of Science and Technology, Manchester, UK, 10–12 April.

De Smedt, K [1990], *Incremental sentence generation: a computermodel of grammatical encoding.* PhD dissertation, University of Nijmegen.

Joshi, A [1987a], An Introduction to Tree Adjoining Grammars. In A. Manaster-Ramer (ed) *Mathematics of Language* John Benjamins, Amsterdam.

Joshi, A [1987b] The relevance of tree adjoining grammar to generation. Chapter 16 in Kempen, G (ed) *Natural Language Generation*, Dordrecht: Martinus Nijhoff.

Kempen, G [1987] A framework for incremental syntactic tree formation. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 23–28.

Kempen, G & Hoenkamp, E [1987] *An Incremental Procedural Grammar for Sentence Formulation.* Cognitive Science April-June.

Kroch, A [1987] Unbounded Dependencies and Subjacency in a Tree Adjoining Grammar In A. Manaster-Ramer (ed) *Mathematics of Language* John Benjamins, Amsterdam.

Kroch, A & Joshi, A [1986] Analyzing Extraposition in a Tree Adjoining Grammar. In G. Huck & A. Ojeda (eds.) *Syntax and Semantics (Discontinuous Constituents)*, Academic Press.

Levelt, W [1989] *Speaking: From Intention to Articulation.* Massachusetts: MIT Press.

Schabes, Y, Abeillé, A & Joshi, A [1988], Parsing Strategies with 'Lexicalized' Grammars: Application to Tree Adjoining Grammars. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest.

Schabes, Y & Joshi, A K [1988], An Earley-type Parsing Algorithm for Tree Adjoining Grammars. In *Proceedings 26th Annual Meeting of the Association for Computational Linguistics.*

Schabes, Y & Joshi, A K [1990], Parsing with Lexicalized TAGs. In M. Tomita (ed.) *Current Issues in Parsing Technology*, Kluwer Academic Press.

Schabes, Y [1990], *Mathematical and Computational Aspects of Lexicalized Grammars.* PhD dissertation, University of Pennsylvania.

Shieber, S [1983], *Direct Parsing of ID/LP Grammars.* In Linguistics and Philosophy 7(2).

Shieber, S [1988] A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics and the 24th Annual Meeting of the Association for Computational Linguistics*, Budapest, Hungary, 22–27 August.

Shieber, S & Schabes, Y [1990a], Synchronous Tree Adjoining Grammars. In *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki.

Shieber, S & Schabes, Y [1990b], Generation and Synchronous Tree-adjoining Grammars. In *Proceedings of the 5th International Workshop on Natural Language Generation*, Pittsburgh.

ITK: P.O. BOX 90153 5000 LE TILBURG THE NETHERLANDS